

## **О Библиотеке Моделирования Процессов**

Библиотека Моделирования Процессов AnyLogic поддерживает дискретно-событийный, или, если быть более точным, "процессный" подход моделирования. С помощью объектов Библиотеки Моделирования Процессов вы можете моделировать системы реального мира, динамика которых представляется как последовательность операций (прибытие, задержка, захват ресурса, разделение, ...) над агентами, представляющими клиентов, документы, звонки, пакеты данных, транспортные средства и т.п. Эти агенты могут обладать определёнными атрибутами, влияющими на процесс их обработки (например, тип звонка, сложность работы) или накапливающими статистику (общее время ожидания, стоимость).

Процессы задаются в форме потоковых диаграмм (блок-схем) - графическом представлении, принятом во многих областях: производстве, бизнес-процессах, центрах обработки звонков, логистике, здравоохранении и т.д. Потоковые диаграммы AnyLogic иерархичны, масштабируемы, расширяемы и объектно-ориентированы, что позволяет пользователю моделировать сложные системы любого уровня детальности. Другой важной особенностью Библиотеки Моделирования Процессов является возможность создания достаточно сложных анимаций процессных моделей.

### **Расширение процессных моделей**

Как и все в AnyLogic, процессная модель может быть расширена до любого требуемого уровня детальности с необходимой нестандартной функциональностью. Во-первых, базовые типы агентов и ресурсов могут быть расширены вашими собственными типами с любыми дополнительными параметрами и методами. Указав эти типы в параметрах объектов библиотеки, вы получите более простой доступ к вашим параметрам. Например, в модели аэропорта вам может понадобиться создать специальные типы, моделирующие пассажиров, багаж, офицеров безопасности и т.д.

Во-вторых, у каждого объекта Библиотеки Моделирования Процессов есть специальные "точки расширения" - места, где вы можете задать какие-то

действия или выражения. Такими точками расширения являются динамические параметры (помеченные в описаниях объектов как *[динамические]*), вычисляемые во время выполнения модели при прохождении агентов через процессную диаграмму. Например, в объекте **Delay**, моделирующем распечатку посадочных талонов в объекте *Регистрация*, вы можете присвоить пассажиру выход (гейт), написав `agent.gate = main.gateof(agent.flightno )` в действии *При выходе*. Здесь `gate` (гейт) и `flightno` (номер полета) - параметр типа агента `Passenger`, моделирующего пассажиров, а `gateof` - функция, заданная в агенте верхнего уровня модели `Main`. Обратите внимание, что у вас есть доступ к любой части модели из любой другой части модели, и более того, поскольку вы пишете Java код, вы можете взаимодействовать с любым внешним программным обеспечением и/или использовать любой из огромного количества Java пакетов.

### **Создание анимаций для процессных моделей**

Библиотека Моделирования Процессов тесно интегрирована с анимационными средствами `AnyLogic` и позволяет создавать анимации процессов любой степени сложности, в том числе иерархические и с несколькими различными графическими представлениями процесса. Например, вы можете определить глобальный взгляд на процесс производства с несколькими агрегированными индикаторами, а также детальные анимации конкретных операций - и переключаться между ними.

Для большинства объектов Библиотеки Моделирования Процессов анимация задается следующим образом: вы рисуете фигуру разметки пространства, скажем, путь, указываете ее в параметре *Место агентов*, например, объекта **Delay** и говорите этому объекту, что вы хотите отображать агентов, задерживающихся в этом объекте, движущимися по этой фигуре анимации. Тогда пока агенты будут находиться в объекте, анимации этих агентов будут отображаться в соответствующих точках этого пути.

## Комбинирование процессных моделей с моделями и конструкциями других типов

Одним из главных преимуществ AnyLogic является возможность комбинирования различных подходов моделирования, позволяющая отражать комплексность и неоднородность систем реального мира. Вы можете комбинировать ваши процессные модели, построенные с помощью Библиотеки Моделирования Процессов, с моделями [системной динамики](#) или [агентными моделями](#), или же просто создавать ваши собственные объекты с помощью базовых элементов AnyLogic и включать их в диаграмму, описывающую ваш процесс.

Есть множество способов комбинирования подходов моделирования, мы приведем лишь некоторые примеры:

- Влияние системно-динамической диаграммы потоков и накопителей на процессную модель может быть реализовано, например, как объект [Source](#), создающий агентов с интенсивностью, управляемой динамической переменной.
- Обратная связь может быть представлена, например, системно-динамическим накопителем, увеличивающим свое значение при каждом прибытии агента в объект процессной диаграммы [Sink](#).
- В определенный момент своего жизненного цикла живущий в популяции агент может вставляться в процесс с помощью блока [Enter](#).

Если вам нужен объект со специфической функциональностью, отличной от функциональности, предоставляемой объектами Библиотеки Моделирования Процессов (и недостижимой путем комбинирования этих объектов), вы можете создать ваш собственный тип агента и использовать объекты [Exit](#) и [Enter](#) в качестве интерфейсных элементов - тогда вы сможете вставлять экземпляры данного типа агентов в вашу процессную диаграмму. В таком агенте можно использовать диаграммы состояний, события, переменные и т.д (см. описание объекта [Exit](#)).

## **Иерархические модели и повторно используемые модельные компоненты**

Если моделируемая система сложна, имеет смысл разбить её модель на компоненты (подпроцессы) и поместить каждый из них в отдельного агента. Вы можете определить входы и выходы из подпроцесса ([порты](#)), поместить их на внешний интерфейс агента и скрыть его реализацию. На верхнем уровне вы будете оперировать такими объектами как блоками, соединяя их входы и выходы. Вы можете создать несколько экземпляров агента с разными параметрами, в том числе и в других проектах.

Пример: вам понадобилось создать в модели аэропорта такие компоненты, как *Регистрация*, *ПаспортныйКонтроль*, *Посадка* и т.д. В блок, моделирующем зону регистрации, вы можете лишь единожды добавить объект *СтойкаРегистрации* и сделать его [реплицированным](#). Ресурсы модели могут разделяться между различными агентами, так что вы можете, например, использовать общий компонент *ИнформационнаяСистема*, предоставляющий ресурсы для этих подпроцессов.